Fair Election of Monitoring Nodes in WSNs

Quentin Monnet, Youcef Hammal, Lynda Mokdad, Jalel Ben-Othman

Fair Election of Monitoring Nodes in WSNs

Quentin MONNET
Lab. LACL, Univ. Paris-Est
LACL (EA 4219), UPEC
F-94010 Créteil, France
quentin.monnet@lacl.fr

Youcef HAMMAL
Lab. LSI, USTHB Univ.
Dept of Computer Science
Algiers, Algeria
hammal@lsi-usthb.dz

Lynda MOKDAD
Lab. LACL, Univ. Paris-Est
LACL (EA 4219), UPEC
F-94010 Créteil, France
lynda.mokdad@u-pec.fr

Jalel BEN-OTHMAN
Lab. L2TI, Univ. Paris 13
L2TI (EA 3043), UP13
F-93430 Villetaneuse, France
jbo@univ-paris13.fr

***Abstract*—In this era of big data, of quantified self and of smart cities, wireless sensor networks are meant to be used every day, for all sort of applications. Made of tiny sensors, they collect data and communicate through wireless technologies. Because they may take part in sensitive or military applications, security is an essential matter in such networks. Confidentiality and authenticity can be ensured by the use of dedicated mechanisms. Focusing on availability, we propose here a new practical approach to protect the network against denial of service attacks thanks to the use of traffic monitoring agents called *cNodes*. The approach uses a fair election process of *cNodes* in accordance with classical criteria related to residual energies and the presence of compromised nodes which may have greedy or jamming behaviors. Results obtained from simulations show that this method is effective both in terms of detection and of energy conservation.**

***Index Terms*—Wireless sensor networks; Reliability, availability, and serviceability; Energy-aware systems**

I. INTRODUCTION

A wireless sensor network (WSN) is a self-organizing and multi-hop network consisting of low-cost and small wireless devices (sensor nodes) that are disseminated over a geographical area for gathering data on some physical phenomena. Delivery of sensory data for process and analysis to a base station is based on a collaborative work of the nodes in a multi-hop fashion.

WSNs may be deployed in hostile or inaccessible environments raising many challenges to ensure that sensors work effectively and survive long enough to fulfil their assignments. Actually, WSN nodes are battery powered and are often required to operate for long periods. Once its battery is exhausted, a node is out of service and network performance is degraded. Wireless transmission and reception make significant demands on available energy in addition to the need to process data, sense the environment *et cætera*. Thus energy efficiency is an important concern in protocol design for multi-hop networks like WSNs and it has strong influence on the design of protocols (to a lesser extent, required computational capabilities and memory are also issues to consider, because both are restricted for sensors that must embed cheap and low-consuming hardware).

Therefore, communications in a WSN shall abide to strict time-constrained rules, as specified by the protocol stack used in the network (such as IEEE 802.15.4™ for instance [1]). When correctly applied, the protocol for the medium access control (MAC) sublayer reduces collisions and improves the transmission rate. However, its correct operating mode is based upon a distributed algorithm that is executed locally on each node to determine the periods of access to the channel. In other terms, it is assumed that all WSN components will abide to the given specifications. Unfortunately, such networks may be implemented in untrusted environments where misbehaving parties can deviate from the protocol specification and achieve better performances at the expense of honest participants, or use jamming techniques which prevent legitimate peers to achieve their communications. Denial of service (DoS) attacks precisely use such changes in these protocol parameters and in the operating mode of some nodes, thereby leading to harmful effects on the overall network performance.

In a previous work [2] an approach to deal with such risks was given. It relies on the use of monitoring nodes (called "*cNodes*") to protect a clustered wireless sensor network against various denial of service attacks. Clustering algorithms are commonly used in an attempt to ease deployment and managing and to better scale wireless sensor networks. They create partitions (clusters) inside the network (see example on Figure 1), each having a cluster head acting as a proxy between the other nodes inside the cluster and the rest of the network. *cNodes* are responsible for listening to the traffic around them and for signaling misbehaving nodes to their cluster heads. They have to remain in listening state all the time they endorse this role. Hence the way they are selected among the nodes can be decisive in regard to detection efficiency as well as load repartition inside clusters. So as to improve this load

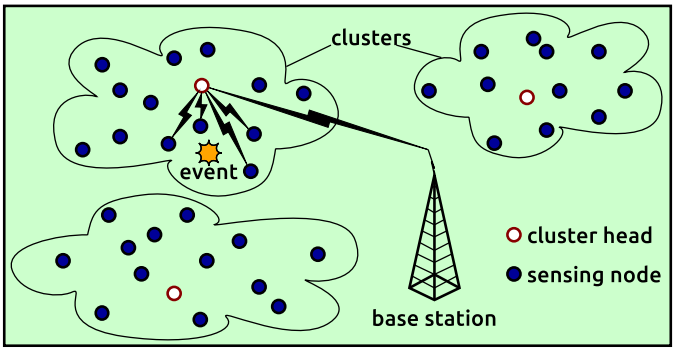

Figure 1. Scheme of an example clustered wireless sensor network

balancing, retained selection process of those control nodes is based on the residual energies of sensors. But as compromised nodes can make false statements about their residual energies, they can easily become *cNodes*, thereby decreasing the odds of being detected. This issue was addressed by introducing a second role of surveillance ("*vNodes*") responsible for watching over the *cNodes* and for matching their announced consumption according to a mathematical model.

However, such a proposal raises a recurrent problem since such *vNodes* themselves have to be selected by the cluster heads according to some criteria that can be faked by compromised nodes. Obviously, it seems inefficient to introduce again a new kind of control role over *vNodes* and to continue with the same problem over and over.

Therefore, we propose in this article a new efficient approach based on a fair election principle. Its purpose is to provide a secure way to designate *cNodes* (*i.e.* to prevent compromised nodes to be selected) while consuming as little energy as possible, and to efficiently balance the load between all nodes of the cluster. Our approach uses an iterative process to always select one kind of control nodes which report to their cluster heads the communications undertaken by their neighboring sensors. In order to detect suspicious and compromised nodes, the cluster heads compute the consumed energy of each node from declared residual energies and then compare it with that assessed according to reported communications from its related *cNodes*. Furthermore, in the first iteration of the process, all nodes are considered as *cNodes* and used for controlling each other in order to eliminate suspicious nodes as much as possible. Each *cNode* is selected henceforth among sane nodes in such a way that each node is controlled by at least two *cNodes*. During next iterations, *cNodes* act as control nodes monitoring surrounding simple sensors as well as other neighboring *cNodes* (in other words they also endorse the role of *vNodes* from formerly proposed approach), thus increasing the likelihood to detect suspicious nodes even though they were selected before as *cNodes*.

The remainder of this paper is organized as follows: Section II presents some related work about discussed topics. We expose our model in Section III, and then the simulation results obtained with the ns-2 tool in Section IV. At last Section V sums up our contribution and brings perspectives for future work.

II. Related work

This section is divided into three parts: security in wireless sensor networks, denial of service specific mechanisms, and clustering algorithms and energy preservation.

A. Security in WSNs

Denial of service is not the only type of attacks a WSN should resist to. Security in general in sensor networks has attracted quite a lot of interest during the last years. Hence it has been the subject of many studies in literature, as well as several state-of-the-art articles [3], [4].

Confidentiality and integrity must be ensured to prevent attackers to access to, or to tamper sensitive data. A number of solutions have been proposed [5], many of them involving strong [6] and/or homomorphic [7] cryptography, some relying on other mechanisms such as multi-path based fragmentation of the packets [8] or game theory [9].

Authentication brings to participants the guarantee that the peer they are communicating with truly is what it claims to be; that is another important point. It has been deeply investigated as well [10]. Many lightweight proposals for key management in WSNs have been suggested [11], [12].

Apart from those, there have been a variety of proposals to secure some other elements, considering that any information about any aspect of the network might be valuable to an attacker. Hence there are approaches, for instance, to secure the geographical location of the nodes through epidemical information dissemination [13] as well as through more conventional mechanisms [14].

B. DoS-specific mechanisms

Denial of service attacks embraces many different attacks, which can target all layers of the network [15]. Jamming the radio frequencies as well as disturbing the routing protocols are just a few examples of ways to degrade the network. In reaction to these, a number of solutions have been proposed [16].

As stated in the introduction, we focus in this paper on inside attackers attempting to bend the MAC protocol parameters to their needs, be it to achieve better performances for themselves (*greedy* attacks) or to generally harm the network (*jamming* attacks or sleep deprivation). To detect such attackers, many solutions rely on trust models [17]. We base our own work on Lai's and Chen's approach [18], which consists in assigning monitoring nodes in the network. Those monitoring nodes, also called *cNodes* in this article, apply a set of rules [19] to overheard traffic so as to detect misbehaving nodes. On multiple rule breaks, they report the suspicious node to the cluster head. To prevent false positive detection, the CH waits for several reports about a given node before considering it as compromised. After that, it virtually excludes the misbehaving node from the cluster by broadcasting a warning to all sensors. We have observed in previous studies that renewing periodically the selection process of those monitoring nodes (*cNodes*) helps saving energy in the network [20], and we have tried ever since to find an optimal selection algorithm to obtain a good equilibrium between security, attack detection and energy preservation [21].

C. Clustering algorithms and energy preservation

A lot of approaches intended to bring security in a WSN are cluster-based [22]. But the main purpose of clustering a sensor network usually resides in scaling possibilities, improved nodes management and energy savings brought by partitioning. Several clustering algorithms have been proposed [23]. They generally aim at determining which nodes in the network will be the cluster heads, often basing the choice on energetic considerations. Basically, choosing a cluster head in a network

is not so different than selecting *cNodes* in a cluster. But in the latter case we have some additional constraints on security.

One of the easiest clustering algorithm to implement, and probably one of the most used is the LEACH algorithm [24]. LEACH makes each node draw a pseudo-random number and matches it with a threshold which was computed from the number of desired cluster heads in the network and from the last iteration when the sensor was selected as a CH. There is a number of proposals derived from LEACH, to improve either its efficiency [25], [26] or its security [27]. Other example clustering algorithms include HEED [28] or FFUCA [29].

Aside from clustering, the importance of energetic issues in WSNs have led to the proposals of several mechanisms to cut down its consumption [30], based for example on packets priority [31].

III. Procedure of fair election of cNodes

Our selection process of *cNodes* is iterative so that new *cNodes* are continually elected and others are discarded because they have already taken this role before or have been declared as suspicious.

Let the symbol $\mathcal{N}$ denote the set of all nodes in the cluster and $\mathcal{CN}$ denote the set of *cNodes*. i is the index ranging over $\mathcal{N}$. Let RE_k be an array of residual energies of nodes reported by *cNodes* to the cluster head at the k^{th} iteration. The symbol $Obs_k[j]$ denotes an array containing observations made by *cNode* j on communications of its neighbors.

The election process starts with an initialization phase and then enters a loop block which is iteratively performed as long as the network is running.

A. Initialization phase

At the start of the process, the following actions are undertaken:

- Each node i of the cluster sends to the cluster head the value of its residual energy, which is stored into a related array $RE_0[i]$.
- Each node acts as a *cNode* and starts controlling its neighbors. It keeps sensing and forwarding data (otherwise there would be no traffic to observe). Since the set of *cNodes* contains all nodes of the cluster, each node starts recording the packets sent by its neighbors.
- The cluster head sets the counter of iterations k to 1.

B. Loop block

At each iteration k, the election process executes the following steps:

1) The duration of this surveillance step at any iteration k is random to prevent compromised nodes to simulate the behavior of sane nodes as long as possible. During step 1, each *cNode* controls the neighboring nodes (including other *cNodes*) by recording and adding up sizes of all packets sent or received by these nodes.
2) At the end of iteration k, the cluster head (CH) asks each node i to send its residual energy value $RE_k[i]$ and asks each *cNode* j to send the array $Obs_k[j]$ containing its observations over transmission rates of its neighbors.
3) For each node i, the cluster head performs an analysis work as follows:
   - According to an adequate mathematical model, the CH assesses the energy consumption ECa related to the maximum of rates $\{Obs_k[j][i]\}|_{j\in\mathcal{CN}}$ observed by neighboring *cNodes* j during the current iteration k.
   - The CH also computes the value of the energy consumption ECd as the difference between residual energies declared in the two last steps (*i.e.* $RE_k[i]$ and $RE_{k-1}[i]$).
   - If $|ECd - ECa| \leq \epsilon$ (where ϵ represents a tolerated error) then the node i is declared as sane and put into the set SEN of nodes eligible to take on a *cNode* functioning mode. Otherwise, it is removed from the set SEN of sane nodes (if it was there) and put into the pool SSN of suspicious nodes.
   - Let $SSN[i]$ stand for the number of times it has declared as suspicious from the start of the process. If $SSN[i] \geq threshold$ then the node is declared as compromised and put into a quarantine list. On the other hand, if a suspicious node has continually been declared as sane (more than some number of times) then it could be removed from SSN and put again into SEN.
4) Once step 3 is finished, the CH selects *cNodes* from the set SEN of eligible nodes in such a way that every node is controlled by at least two *cNodes*. Hence, as a *cNode* will be controlled by other *cNodes*, its misbehavior would be reported to the CH. For further iterations, such a rule helps detect and discard compromised nodes which have been chosen as *cNode* because they normally behaved in the past iterations and made false statement about their residual energies, unless they continue to undertake normal communications.
5) The process increments the number k of iterations and continues its iterative execution by going back to step 1.

C. Mathematical model for energy consumption

A possible mathematical model that the cluster heads may use for computing the energy consumed by the nodes based on received observations is Rakhmatov and Vrudhula's diffusion model [32]. It provides a pretty accurate approximation of real consumption, taking into account chemical processes internal to the battery such as rate capacity effect and recovery effect. Rakhmatov and Vrudhula's diffusion model refers to the chemical reaction happening inside the battery electrolyte, and is summarized by equation (1):

$$\sigma(t) = \underbrace{\int_0^t i(\tau)\,\mathrm{d}\tau}_{l(t)} + \overbrace{\int_0^t i(\tau)\left(2\sum_{m=1}^{\infty}\exp^{-\beta^2 m^2(t-r)}\right)\mathrm{d}\tau}^{u(t)} \tag{1}$$

where:

- $\sigma(t)$ is the apparent charge lost from the battery at t.
- $l(t)$ is the charge lost to the load ("useful" charge).
- $u(t)$ is the unavailable charge ("lost in battery" charge).
- $i(t)$ is the current at t.
- $\beta = \frac{\pi\sqrt{D}}{w}$, where D is the diffusion constant and w the full width of the electrolyte of the battery.

In practice, computing the first ten terms of the sum provides a good approximation.

D. Selecting cNodes among set SEN

At step 3 of the fair election process, cluster heads select the *cNodes* for the running iteration among the nodes inside SEN sets. Note that any selection criterion could be used at this point. For instance, *cNodes* could be randomly picked among elements of SEN until we have:

- enough *cNodes* (according to user's choice)
- and all nodes covered by at least two *cNodes*, as mentioned above.

Some other selection criteria could include:

- residual energy of the nodes
- connectivity index (number of direct neighbors)
- signal power
- *et cætera*

Several criteria can even be combined to obtain a weighted score, such as for instance in equation 2:

$$s_k[i] = (\alpha \times RE_k[i]) + (\gamma \times ci_k[i]) + (\delta \times sp_k[i]) + (\zeta \times nsl_k[i]) \quad (2)$$

where:

- $s_k[i]$ denotes the score for node i at iteration k.
- $RE_k[i]$ remains the residual energy for said node and iteration.
- $ci_k[i]$ would be the connectivity index of i.
- $sp_k[i]$ is the average signal power as perceived by the neighbors of node i.
- $nsl_k[i]$ is the maximum value between a predetermined integer value and the number of iterations before k since node i was last selected as a *cNode*.
- α, γ, δ and ζ would be constants fixed by the user.

This formula could be used to sort nodes in set SEN so that the cluster head can select the best possible *cNodes* in regard to retained criteria.

Setting the criteria and the weights for the formula is up to the user of the network. It should be adapted to the exact application and environment of the WSN. For instance it may be worth noting that for networks made of static nodes, the connectivity index and the signal power of the nodes are not expected to change much between two consecutive iterations. Therefore their weights should not be too high (in regard to the other weights in the formula) so as to avoid selecting the same *cNodes* at each iteration. In clustered networks, where all the nodes can reach their CH in a one-hop fashion, index connectivity or signal power might not even be relevant (once again, depending also on the deployed application). But even if we only work in clustered networks in this study — because they allow energy savings and a much better scaling — we should nevertheless consider other architectures in that respect, for not all WSNs are clustered. And many ones also work with mobile nodes. In such cases, evaluating the density of nodes or the quality of the links in the area where the candidate *cNodes* are located becomes more interesting because it generally has a higher impact on performances. Such observations remain valid for the chosen constraint stating that each node must be watched over by at least two *cNodes*. While it is a good thing in clusters where nodes are all gathered around their cluster head, it could be much harder to obtain in some other topologies. In a star-like network for instance, where most sensors would be on branches and only have two neighbors (one closer, one farther from the base station), it could result in all nodes being selected as *cNodes*.

IV. Simulation results

So as to compare our proposal with previous approaches, we have undertaken simulations using ns-2 software with parameters displayed in Table I. A star (*) denotes a value used only for compromised node. To play their role during the whole duration of the simulation, the compromised node as well as the cluster head were granted more energy than the normal sensors. The simulated cluster is a grid with ten sensors on each side, and the transmission range is such that the nodes in the corner can just reach until the cluster head, as displayed on Figure 2. We compared the fair election process with a (pseudo-)random selection such as in [20].

Table I
Simulation parameters

Parameter	Value
Simulation time	3,600 seconds
Initialization phase	30 seconds
Number of sensors	100 (+ cluster head)
Topology	regular grid (72m×72m)
Compromised node(s) number	1
cNodes percentage	7 to 10 %
Transmission rate	1 kbits/s — 35 kbits/s*
Transmission range	50 meters
Packets size	500 bytes
Reception consumption	0.395 W
Emission consumption	0.660 W
Initial energy amount	10 J — ∞*

A. Detection rate

The number of *cNodes* which detected the compromised node at each iteration are reported on Figure 3. After 2,000 seconds, nearly all nodes in the cluster are dead. Before that, we can observe that the efficiency of the two methods, with 10 % of *cNodes*, is nearly the same. The random selection process can afford monitoring a little longer but with a lower detection rate; the fair election process ensures that all nodes are monitored by at least two *cNodes*, therefore the number of

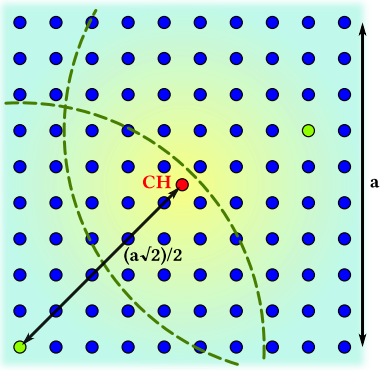

Figure 2. Topology of simulated cluster; transmission range is $(a\sqrt{2})/2$

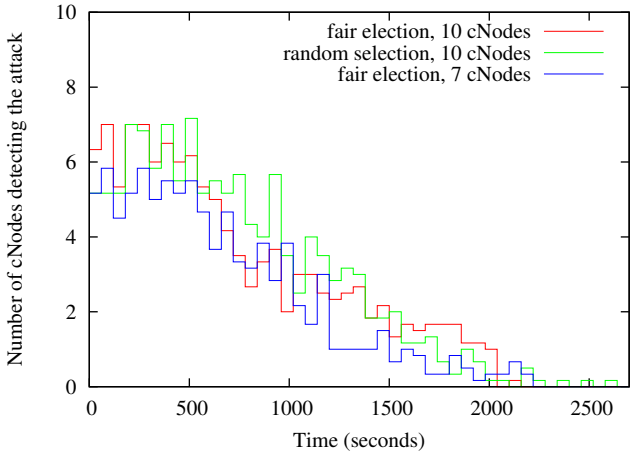

Figure 3. Denial of service attack detection

cNodes detecting the compromised node takes longer to drop under 2.

We tried to see what results would give another instance of the fair election process, but with only seven *cNodes*. In this case the average detection rate is slightly lower (since we have less *cNodes*), but the detection is still performed by at least one *cNode* after 2,200 seconds. With the random selection process, the use of seven *cNodes* only drops significantly the efficiency of the detection method. It was not displayed on the graph.

B. Energy

Figure 4 presents the time of death of the normal sensors in the cluster. As expected, the fair election process leads to a faster exhaustion of the battery. This is due in part to the initialization phase, when all nodes remain in a listening state all along; and also to the emission of the observations of the nodes to the cluster head at each iteration. The difference between the two methods is quite low: the initialization phase plays for most of it, and it would be softened over time with real batteries (remember that we worked with 10 J only).

The test involving seven *cNodes* gives better results than both methods using ten monitoring nodes. We can observe

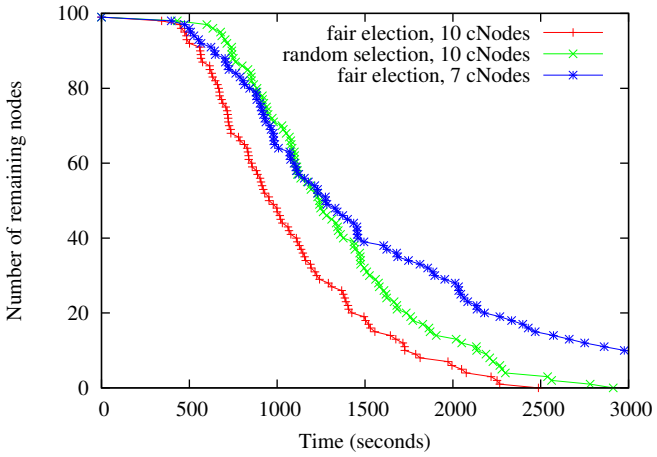

Figure 4. Time of death of the nodes

an early drop due to the increased consumption of the initial phase, but after 1,300 seconds there are more nodes alive than with the random selection process used to pick ten *cNodes*. The fewer *cNodes* the cluster have, the less indeed they consume energy.

C. Compromised cNodes

Using only seven nodes with the fair election process seems to be a good approach. The detection rate remains correct, while the energy consumption is lower than other methods using more monitoring nodes. But with fewer *cNodes* there is a risk that compromised nodes manage to monopolize the roles, and that nothing remains to detect them.

The observations submitted to the cluster head and the constraint we used to enforce that each node be monitored by at least two *cNodes* are useful here. With a random selection process (*i.e.* without applying those constraints) we had the compromised node elected as a *cNode* during 12 % of the simulation time on average (it is a little more than the theoretical 10 % expected with 10 % of *cNodes* in the cluster). The fair election process and its additional constraint were enough, within the conditions of our simulations, to prevent completely the compromised node to be selected as a *cNode* at any time.

V. Conclusion and Perspectives

Monitoring nodes (*cNodes*) can be used in WSNs so as to detect compromised nodes and to virtually exclude them. This role consumes more energy than normal sensing, and for obvious security reasons it must not be assigned to rogue sensors. In this article we proposed a novel approach to select those *cNodes* based on an iterative fair election principle for clustered wireless sensor networks. After an initialization phase and for each cluster, the cluster head establishes a list of eligible nodes from the observations of the sensors relative to their neighbors, and then selects the *cNodes* using the best suited criteria for the application run in the network.

Experiments performed through the ns-2 simulator show that even if global consumption is slightly higher than for a selection process based on randomness, the detection rate is good, and the method can even afford to use less *cNodes* without loosing in efficiency. In addition to this, the risk of selecting a compromised node as *cNode* is drastically reduced.

Based on this work, we would like to push our experiments in further study so as to analyze the impact of our solution on networks with different topologies (for instance, non-clustered networks or networks with areas of different activity levels), and to observe the consequences of the different parameters used by the cluster head to proceed to the selection itself. Additional perspectives include improving the security of this approach by monitoring the cluster head, as well as formal modeling of the solution and validation through model-checking tools.

REFERENCES

[1] "IEEE Standard 802.15.4™ (2003) Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Standard for Information technology, IEEE-SA Standards Board.

[2] Y. Hammal, Q. Monnet, J. Ben-Othman, L. Mokdad, and A. Abdelli, "Timed automata based modeling and verification of denial of service attacks in wireless sensor networks," *Studia Informatica Universalis*, vol. 12, 2014.

[3] A. Y. Dak, S. Yahya, and M. Kassim, "A literature survey on security challenges in VANETs," *International Journal of Computer Theory and Engineering*, vol. 4, no. 6, pp. 1007–1010, Dec. 2012.

[4] S. Alam and D. De, "Analysis of security threats in wireless sensor network," *International Journal of Wireless and Mobile Networks*, vol. 6, no. 2, pp. 35–46, Apr. 2014.

[5] S. Ozdemir and Y. Xiao, "Secure data aggregation in wireless sensor networks: a comprehensive overview," *Computer Networks*, vol. 53, no. 12, pp. 2022–2037, Aug. 2009.

[6] M. A. Simplicio, Jr, B. T. de Oliveira, P. S. L. M. Barreto, C. B. Margi, T. C. M. B. Carvalho, and M. Naslund, "Comparison of authenticated-encryption schemes in wireless sensor networks," in *Proceedings of the 36th Annual IEEE Conference on Local Computer Networks*, Bonn, Germany, Oct. 2011, pp. 454–461.

[7] S. Ben Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and integrity for data aggregation in WSN using homomorphic encryption," *Wireless Personal Communications*, Sep. 2014.

[8] Q. Monnet, L. Mokdad, and J. Ben-Othman, "Data protection in multipaths WSNs," in *Proceedings of the eighteenth IEEE Symposium on Computers and Communications (ISCC'13)*, Split, Croatia, Jul. 2013.

[9] H.-Y. Shi, W.-L. Wang, N.-M. Kwok, and S.-Y. Chen, "Game theory for wireless sensor networks: a survey," *Sensors*, vol. 12, no. 7, pp. 9055–9097, Jul. 2012.

[10] P. Guo, J. Wang, J. Zhu, and Y. Cheng, "Authentication mechanism on wireless sensor networks: A survey," in *Proceedings of the 2nd International Conference on Information Technology and Computer Science (ITCS'13)*, vol. 25, Beijing, China, Jul. 2013, pp. 425–431.

[11] P. Guo, J. Wang, J. Zhu, Y. Cheng, and J.-U. Kim, "Construction of trusted wireless sensor networks with lightweight bilateral authentication," *International Journal of Security and Its Applications*, vol. 7, no. 5, pp. 225–236, Sep. 2013.

[12] H. Bawa, P. Singh, and R. Kumar, "An efficient novel key management scheme for enhancing user authentication in a WSN," *International Journal of Computer Network and Information Security*, vol. 5, no. 1, pp. 56–64, Jan. 2013.

[13] L. Kazatzopoulos, C. Delakouridis, and C. Anagnostopoulos, "WSN location privacy scheme enhancement through epidemical information dissemination," *International Journal of Communication Networks and Information Security*, vol. 6, no. 2, pp. 162–167, Aug. 2014.

[14] C. M. George and M. Kumar, "Cluster based location privacy in wireless sensor networks against a universal adversary," in *Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES'13)*, Chennai, India, Feb. 2013, pp. 288–293.

[15] A. Varshovi and B. Sadeghiyan, "Ontological classification of network denial of service attacks: basis for a unified detection framework," *Scientia Iranica*, vol. 17, no. 2, pp. 133–148, Dec. 2010.

[16] S. K. Singh, M. P. Singh, and D. K. Singh, "A survey on network security and attack defense mechanism for wireless sensor networks," *International Journal of Computer Trends and Technology*, May 2011.

[17] M. Momani and S. Challa, "Survey of trust models in different network domains," *International Journal of Ad Hoc, Sensor and Ubiquitous Computing*, vol. 1, no. 3, pp. 1–19, Sep. 2010.

[18] G. H. Lai and C.-M. Chen, "Detecting denial of service attacks in sensor networks," *Journal of Computers*, vol. 4, no. 18, Jan. 2008.

[19] M. R. Rohbanian, M. R. Kharazmi, A. Keshavarz-Haddad, and M. Keshtgary, "Watchdog-LEACH: a new method based on LEACH protocol to secure clustered wireless sensor networks," *Advances in Computer Science: an International Journal*, vol. 2, no. 3, pp. 105–117, Jul. 2013.

[20] P. Ballarini, L. Mokdad, and Q. Monnet, "Modeling tools for detecting DoS attacks in WSNs," *Security and Communication Networks*, vol. 6, no. 4, pp. 420–436, Apr. 2013.

[21] Q. Monnet, L. Mokdad, and J. Ben-Othman, "Energy-balancing method to detect denial of service attacks in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC'14)*, Sydney, NSW, Australia, Jun. 2014.

[22] A. Ghosal and S. DasBit, "A lightweight security scheme for query processing in clustered wireless sensor networks," *Computers and Electrical Engineering*, Apr. 2014.

[23] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14–15, pp. 2826–2841, Oct. 2007.

[24] M. J. Handy, M. Haase, and D. Timmerman, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Proceedings of the 4th IEEE International Workshop on Mobile and Wireless Communications Networks*, Stockholm, Sweden, 2002, pp. 368–372.

[25] B. B. Reddy and K. K. Rao, "A modifiedclustering for LEACH algorithm in WSN," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 5, pp. 79–83, May 2013.

[26] N. Chawla and A. Jasuja, "Algorithm for optimizing first node die (FND) time in LEACH protocol," *International Journal of Current Engineering and Technology*, vol. 4, no. 4, pp. 2748–2750, Aug. 2014.

[27] L. B. Oliveira, A. Ferreira, M. A. Vilaça, H. C. Wong, M. Bern, R. Dahab, and A. A. F. Loureiro, "SecLEACH — on the security of clustered sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 2882–2895, Dec. 2007.

[28] O. Younis and S. Fahmy, "HEED: a Hybrid, Energy-Efficient Distributed clustering approach for ad-hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

[29] S. Fouchal, Q. Monnet, D. Mansouri, L. Mokdad, and M. Ioualalen, "A new clustering algorithm for wireless sensor networks," in *Proceedings of the seventeenth IEEE Symposium on Computers and Communications (ISCC'12)*, Nevşehir, Turkey, Jul. 2012.

[30] G. Anastasi, M. Conti, M. di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[31] P. Sivakumar, K. Amirthavalli, and M. Senthil, "Power conservation and security enhancement in wireless sensor networks: A priority based approach," *International Journal of Distributed Sensor Networks*, May 2014.

[32] D. Rakhmatov and S. Vrudhula, "An analytical high-level battery model for use in energy management of portable electronic systems," in *Proceedings of the International Conference on Computer Aided Design (ICCAD'01)*, San Jose, CA, USA, Nov. 2001, pp. 488–493.